

NFP107**EXAMEN - 1^{ère} session – Juin 2016****(Tout document écrit autorisé – Téléphones et ordinateurs portables interdits)****Exercice 1 : partie modélisation (5,50 pts)**

On désire modéliser une base de données permettant de mémoriser les informations liées à l'évaluation de livres par des internautes. Tous les mois, un internaute peut évaluer jusqu'à 4 livres. Un livre est caractérisé par son titre, ses auteurs et son édition. L'édition est caractérisée par un format et un code d'éditeur. La relation entre un livre et une édition est caractérisée par un prix de vente et une date correspondant à l'année d'édition du livre. Par exemple un livre peut exister en plusieurs éditions : livre de poche, livre compact, etc.

Un auteur est caractérisé par son nom, son prénom et sa date de naissance. Un internaute est caractérisé par son nom, son prénom, sa date de naissance, son adresse mail et un mot de passe. Un livre possède un numéro ISBN unique qui diffère selon l'édition mais qui est toujours renseigné. Un internaute possède une adresse mail qui est unique et toujours renseignée.

Chaque évaluateur note un livre en tenant compte de l'édition, car deux livres dans deux éditions différentes peuvent être perçus de façon différente par le lecteur.

À chaque évaluation d'un livre dans une édition particulière, la note de l'internaute et la date de l'évaluation sont mémorisées. Un évaluateur ne peut évaluer un livre dans une édition particulière qu'une seule et unique fois.

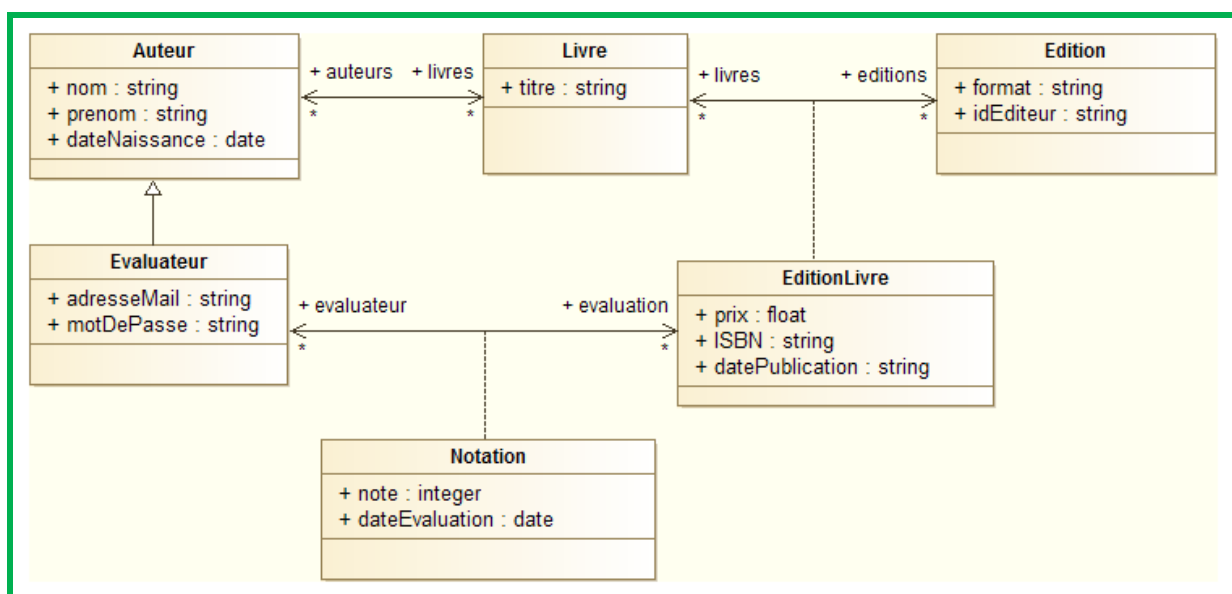
Attention : L'évaluation d'un livre dans plusieurs formats est considérée comme plusieurs évaluations. Cette évaluation est représentée par une note entre 0 et 5.

Remarque : On veillera à minimiser la redondance d'information.

Un évaluateur peut aussi être un auteur.

Vous ajouterez tous les identifiants que vous jugerez nécessaire d'introduire.

- 1) Réalisez le diagramme de classes le plus complet possible correspondant à cette base de données. (1,50 pt)



- 0,50 pt si le schéma est factorisé et que la redondance d'information est minimisée
- 0,25 pt si les cardinalités sont correctes
- 0,25 pt pour les classes d'association
- 0,50 pt pour le schéma

2) Traduisez le diagramme de classes de la question précédente selon le modèle relationnel. Vous introduirez les identifiants que vous jugerez nécessaires **(2,25 pts)**

- | | | |
|----------------|---|------------------|
| • AUTEUR | (<u>IDA</u> , NOM, PRENOM, DDN) | (0,25 pt) |
| • EVALUATEUR | (# <u>IDA</u> , ADRESSEMAIL, MOTDEPASSE) | (0,50 pt) |
| • LIVRE | (<u>IDL</u> , TITRE) | (0,25 pt) |
| • EDITION | (<u>IDE</u> , FORMAT, CODEEDITEUR) | (0,25 pt) |
| • EDITIONLIVRE | (# <u>IDL</u> , # <u>IDE</u> , PRIX, ISBN, DATEPUBLI) | (0,50 pt) |
| • NOTATION | (# <u>IDL</u> , # <u>IDE</u> , # <u>IDA</u> , NOTE, DATEEVAL) | (0,50 pt) |

Compter juste aussi si l'auditeur utilise ISBN en lieu et place de (IDL, IDE) dans la relation NOTATION.

3) Existe-t-il dans ces relations des clefs candidates ? **(0,50 pt)**

Oui dans :

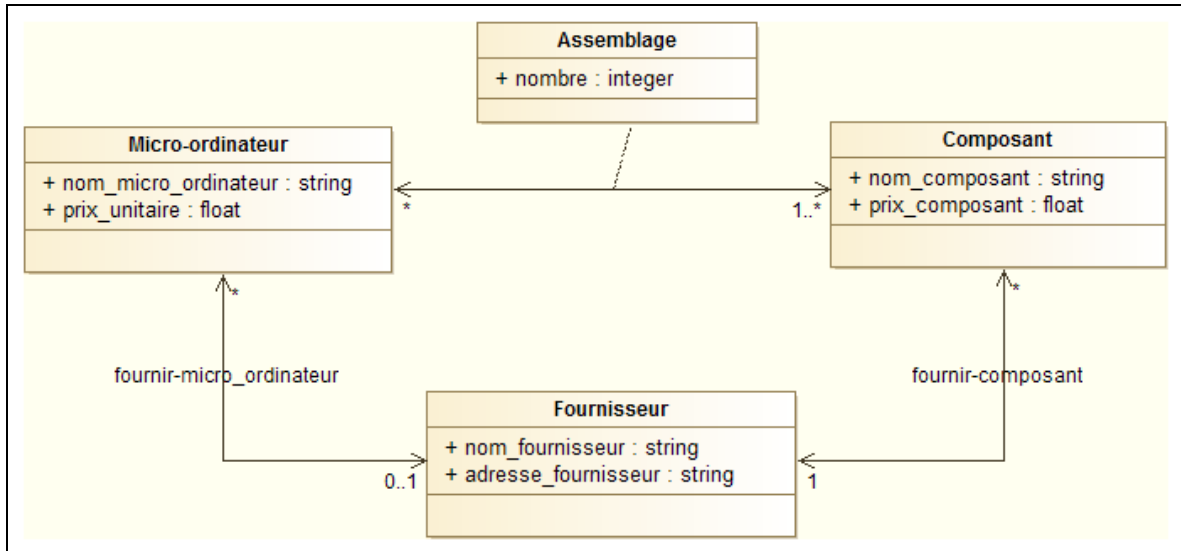
- NOTATION, ISBN est une clef candidate car elle permet d'identifier un livre dans une édition particulière
- EVALUATEUR, ADRESSEMAIL est une clef candidate car elle unique pour chaque évaluateur et toujours renseignée.

4) Donnez les contraintes présentes dans l'énoncé et le diagramme de classes qui devraient être implantées d'une manière ou d'une autre lors de la mise en œuvre de ce modèle pour avoir une mise en œuvre la plus proche possible de la réalité. S'il existe des contraintes de clef candidate il faudra aussi les indiquer. Attention, l'utilisation de SQL n'est pas demandée, seule une description textuelle l'est **(1,25 pt)**

- La note est comprise entre 0 et 5 ; **(0,25 pt)**
- L'attribut IDA de notation est une clef étrangère de la relation EVALUATEUR ; **(0,25 pt)**
- Le prix est un nombre positif ou nul. **(0,25 pt)**
- Les attributs ADRESSEMAIL et ISBN sont toujours renseignés et unique. **(0,25 pt)**
- Un évaluateur ne peut pas évaluer plus de 4 livres par mois. **(0,25 pt)**

Exercice 2 : partie SQL (14,50 pts)

On se propose d'implanter le diagramme de classes suivant qui permet de gérer une société qui commercialise des micro-ordinateurs qu'elle assemble à partir de composants ou qu'elle achète pour les revendre en l'état. Elle se procure les composants et/ou les micro-ordinateurs auprès de divers fournisseurs.



Le modèle relationnel correspondant au diagramme ci-dessus est le suivant :

- MICRO (CM, NOMM, PUM, #CF)
- COMPO (CC, NOMC, PUC, #CF)
- FOURN (CF, NOMF, ADRF)
- ASS {#CM, #CC, NBRE}

Formats des rubriques :

- ☑ CM, CC, CF : 5 caractères
- ☑ NOMM, NOMC, NOMF : 30 caractères maximum
- ☑ ADRF : 100 caractères maximum
- ☑ PUM, PUC, NBRE : nombres

Remarques :

- ☑ Pour chaque réponse, vous veillerez à fournir une solution optimale ;
- ☑ Les 2 premiers caractères du code ordinateur représentent son type, qui peut être : **AC** (pour acheté) ou **AS** (pour assemblé) ;
- ☑ NBRE représente le nombre d'occurrences d'une même pièce dans un ordinateur. Par exemple, deux barrettes mémoires de 2GO.
- ☑ **Les ordinateurs assemblés n'ont jamais de fournisseur ;**
- ☑ **Les ordinateurs achetés ont forcément un fournisseur ;**
- ☑ **La seule syntaxe acceptée est celle d'Oracle.**

I. Compréhension du schéma (3,75 pts)

- 1) Proposez un exemple **pertinent** couvrant les différents cas possibles pour cette base de données en fournissant quelques enregistrements pour les différentes relations. **(0,75 pt)**

MICRO	CM	NOMM	PUM	CF
	AC001	Portable top niveau	2 000,00	F0001
	AS001	The Gamer	1 500,00	

COMPO	CC	NOMC	PUC	CF
	C0001	I7 intel	400,00	F0002
	C0002	RAM Kingston 2GO	300,00	F0002
	C0003	Western digital 1TO	200,00	F0003

FOURN	CF	NOMF	ADRF
	F0001	FNAC	Paris
	F0002	RSM	Toulouse
	F0003	Micro-diffusion	Toulouse

ASS	CM	CC	NBRE
	AS001	C0001	1
	AS001	C0002	2
	AS001	C0003	1

- 2) Implantez la(les) contrainte(s) "not null" représentée(s) sur le schéma conceptuel **(0,50 pt)**

alter table COMPO

modify CF constraint NN_COMPO_CF not null ;

- 3) Implantez la contrainte qui exprime que le code d'un ordinateur commence soit par AC soit par AS **(0,50 pt)**

alter table MICRO

add constraint CHK_MICRO_CM check (substr(CM, 1, 2) in ('AC', 'AS')) ;

- 4) Implantez la contrainte qui dit que si un ordinateur est assemblé alors son code fournisseur est null tandis que s'il est acheté alors son code fournisseur est forcément renseigné (on pourra scinder la contrainte en deux sous-contraintes) ? **(1,00 pt)**

alter table MICRO

**add constraint CHK_MICRO_CF_1 check ((CM not like 'AC%' or CF is null)
and (CM not like 'AS%' or CF is not null)) ;**

- 5) Implantez la contrainte exprimant que tous les micro-ordinateurs ont un nom différent **(0,50 pt)**

alter table MICRO

add constraint UNIQ_MICRO_NOMM unique(NOMM) ;

6) En considérant que toutes les contraintes de clefs primaires sont correctement positionnées, implantez les contraintes de clef étrangère des différentes relations ? **(0,50 pt)**

alter table MICRO

add constraint FK_MICRO_CF foreign key (CF) references FOURN(CF) ;

alter table COMPO

add constraint FK_COMPO_CF foreign key (CF) references FOURN(CF) ;

alter table ASS

add constraint FK_ASS_CM foreign key (CM) references MICRO(CM) ;

alter table ASS

add constraint FK_ASS_CC foreign key (CC) references COMPO(CC) ;

II. Donnez les requêtes suivantes dans le langage SQL (10,75 pts)

7) Donnez l'identifiant des fournisseurs qui ne fournissent pas de micro-ordinateur (**not in, not exists, minus et demi-jointure**). Pour la requête avec le « not in », vous expliquerez clairement pourquoi il faut utiliser une condition « is not null » ou pourquoi elle est inutile **(2,75 pts)**

select CF

from FOURN

(ou COMPO)

where FOURN.CF not in (

select CF

from MICRO

where CF is not null)

(0,50 pt)

La clause « CF is not null » est nécessaire car si le micro-ordinateur est assemblé il n'a pas de fournisseur et donc CF est null. **(0,25 pt)**

select CF

from FOURN

(ou COMPO)

where not exists (select *

from MICRO

where MICRO.CF = FOURN.CF)

(0,75 pt)

select CF

from FOURN

(ou COMPO)

minus

select CF

from MICRO ;

(0,50 pt)

select distinct FOURN.CF

from FOURN, MICRO

(ou COMPO)

where FOURN.CF = MICRO.CF (+)

and MICRO.CM is null ;

(0,75 pt)

8) Donnez le nombre moyen de pièces différentes constituant un ordinateur assemblé **(1,00 pt)**

```
select avg(count(*)) from ASS  
group by CM ;
```

9) Donnez le code et le nom des fournisseurs qui fournissent à la fois des micro-ordinateurs et des composants **(1,00 pt)**

```
select CF, NOMF from FOURN  
where CF in ( select CF from MICRO  
             intersect  
             select CF from COMPO) ;
```

ou

```
select distinct CF, NOMF from FOURN, COMPO, MICRO  
where FOURN.CF = COMPO.CF  
and FOURN.CF= MICRO.CF ;
```

(Il y a d'autres solutions)

10) Ecrivez la requête permettant d'extraire la liste des composants (nom et nombre d'occurrences des pièces) du micro-ordinateur de nom « ***the gamer*** ». ***Vous donnerez en plus de la version SQL la version en algèbre relationnelle.*** Vous veillerez à optimiser l'expression en algèbre relationnelle. **(2,00 pts)**

```
select COMPO.NOMC, ASS.NBE  
from COMPO, ASS, MICRO  
where MICRO.CM = ASS.CM  
and COMPO.CC = ASS.CC  
and MICRO.NOMM = "the gamer" ;
```

(1,00 pt)

